

Password Strength Analyzer & Custom Wordlist Generator

Abstract

This project demonstrates a practical cybersecurity tool developed in Python that analyzes password strength and creates custom wordlists from user-supplied data such as names, dates, or pet names. Using the `zxcvbn` library, the tool estimates password entropy, gives feedback on strength, and generates realistic password candidates through common transformations like leetspeak, case variations, and numeric suffixes. This helps illustrate how attackers form guesses and how to select stronger passwords.

Introduction

Weak passwords are one of the most exploited vulnerabilities in digital systems. Users often reuse predictable patterns, personal details, or common sequences, making them easy to crack. The Password Strength Analyzer and Custom Wordlist Generator tool was designed to evaluate password robustness and simulate potential attack wordlists based on user data. The analysis offers insight into password security and practical recommendations for improvement.

Tools Used

- Python 3.x – Core language for development
- `zxcvbn` – Library for estimating password strength and entropy
- `itertools`, `os` – For transformations and wordlist generation
- `tkinter` – Optional GUI interface for ease of use
- `reportlab` – Used to generate this PDF report

Steps Involved in Building the Project

1. Define project objectives: password analysis and secure wordlist creation.
2. Implement entropy calculations using `zxcvbn` to measure strength.
3. Accept user inputs like name, year, and pet to personalize the list.
4. Apply transformations such as leetspeak (a→@, o→0), capitalization, and year appends.
5. Export results to a `.txt` file (`custom_wordlist.txt`) for penetration testing tools.
6. Add user interface for accessibility, and produce visual feedback on password quality.

Password Strength Analysis (Sample Output)

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Enter a password to analyze (won't be stored): python@123
Enter words for wordlist (e.g. name pet year), separated by spaces: 12 4 2005

--- Password Strength Analysis ---
Password (analyzed): python@123
Score: 2 / 4
Estimated guesses: 27090000
Crack time estimates (display):
  online_throttling_100_per_hour: 30 years
  online_no_throttling_10_per_second: 1 month
  offline_slow_hashing_1e4_per_second: 45 minutes
  offline_fast_hashing_1e10_per_second: less than a second
```

Custom Wordlist Generation (Sample Output)

```
--- Generating Wordlist ---
Base words count (after basic transforms): 33
✅ Wordlist generated: custom_wordlist.txt
Total entries written: 613

Sample (first 20 entries):
01. 12
02. 12!
03. 12!007
04. 12!1234
05. 12!1990
06. 12!1995
07. 12!2000
08. 12!2005
09. 12!2010
10. 12!2015
11. 12!2020
12. 12!2025
13. 12007
14. 1201
15. 1201007
16. 12011234
17. 12011990
18. 12011995
19. 12012000
20. 12012005
```

Conclusion

The Password Strength Analyzer with Custom Wordlist Generator is a powerful educational tool demonstrating how passwords can be tested and improved. It bridges the gap between user behavior and security practice by showing how small variations in passwords affect resistance to attacks. Future enhancements could include cloud integration, real-time feedback systems, and advanced pattern detection using machine learning techniques.

Generated Report | Password Strength Analyzer Project | Use responsibly for ethical cybersecurity testing.